GHULAM ISHAQ KHAN INSTITUTE OF ENGINEERING SCIENCES AND TECHNOLOGY • PAKISTAN •

Abdul Wasay

2023908

CS327 – GPU PROGRAMMING

Faculty of Computer Science and Engineering

BSAI

Task 1: GPU Properties

The properties output matched expectations.

Output below:

```
Device Number: 0
  Device name: NVIDIA GeForce MX130
  Memory Clock Rate (KHz): N/A (Unknown in this toolkit)
  Memory Bus Width (bits): N/A (Unknown in this toolkit)
  Peak Memory Bandwidth (GB/s): ~40.1
  Clock Rate (KHz): N/A (Unknown in this toolkit)
  Compute capability: 5.0
  Multiprocessor count: 3
  Estimated Cores per SM: 128 (Total Cores: 384)
  Peak Compute Performance (GFLOPs): 861.695984
```

Task 2: CPU Matrix Multiplication

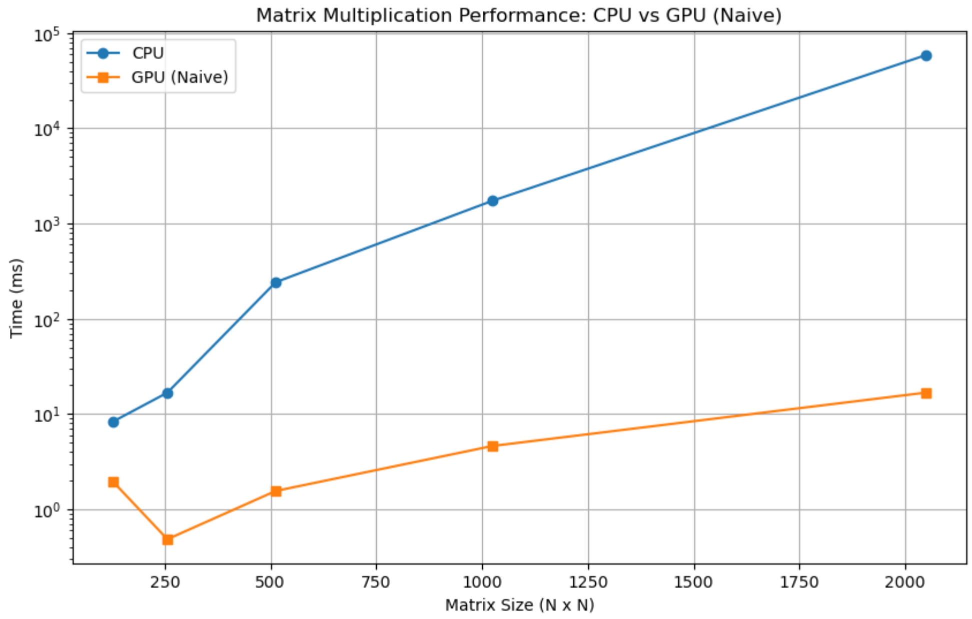
Matrix multiplication implemented in C++. Output format: Two matrices of size rA x cA and rA x cB space-separated floats.

Task 3: GPU Matrix Multiplication (Naive)

Matrix multiplication logic ported to CUDA. To calculate computational intensity:
Computational Intensity = (Total FLOPS) / (Total Memory Access in Bytes)
Memory Bytes = (N*N + N*N + N*N) * 4 bytes = 12 * N^2 bytes.
Floating Point Operations = N^3 multiplications + (N-1)N^2 additions ≈ 2*N^3 FLOPS.
Intensity = 2*N^3 / (12 * N^2) = N / 6 FLOPS/byte

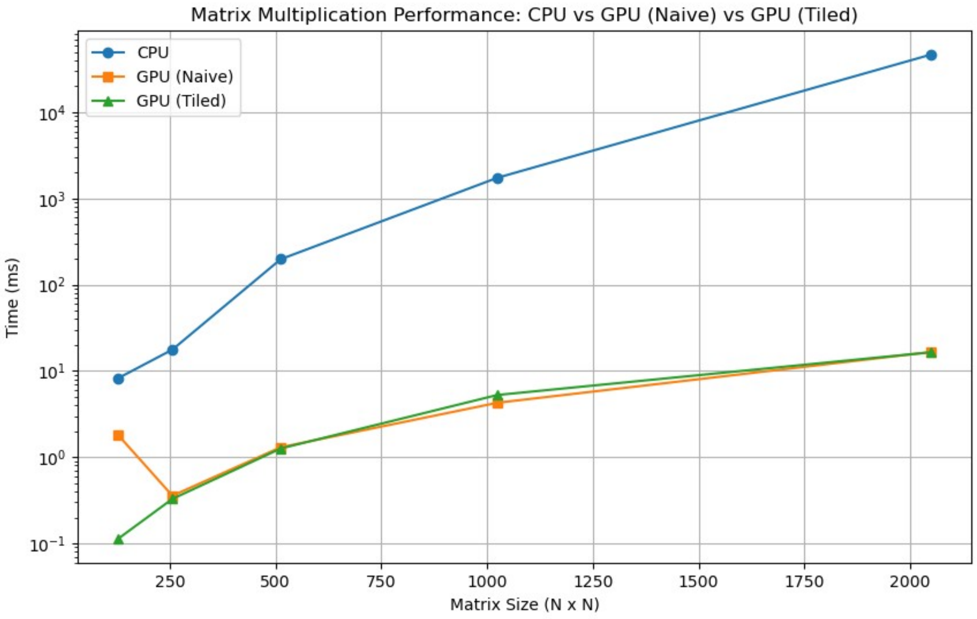
Task 4: Measure Computation Time

Below is the visualization of the naive CPU and GPU execution times measured.



Task 5: Improve Performance with Tiling

The maximum tile size that fits across architectures is typically 32x32, allowing all 1024 threads to leverage 48KB shared memory effectively.
Computational Intensity = 2*N^3 / ((2*N^3 / 32) * 4) = 16 FLOPS/byte.
Below is a plot comparing CPU execution time, GPU Naive Execution time, and GPU Tiled Execution time for different sizes of matrix dimensions.


Matrix Multiplication Performance: CPU vs GPU (Naive) vs GPU (Tiled)
CPU
GPU (Naive)
GPU (Tiled)
Time (ms)
10^4
10^3
10^2
10^1
10^0
10^{-1}
Matrix Size (N x N)